Jeffrey Reid Module 1.3 January 12, 2025

The History of DevOps

DevOps began as a methodology to solve the complications of Development teams working with Operations teams together. Shortly after the switch from the Waterfall development method to the Agile Method began, new issues started to become prevalent. At the time, all work developing the code was completely separate from work fixing or upgrading it. This included problems like different leadership, different goals, and working in entirely different areas. Communication was very lacking and served only to slow down progress and create more problems. In 2007, a fellow named Patrick Debois started the DevOps movement. The following year he joined an Agile Conference started by Andrew Shafer to attempt to devise a solution to the problems he had been having. It wasn’t until 2009 though when the DevOps idea began to take off after a talk at the Velocity Conference. At this point, the idea of the formerly separate Development and Operations Teams working together began to bear fruit. It was over the next several years that this practice spread through various means. This included the first DevOpsDays conference held by Patrick as well as the release of the Continuous Development book. One of the biggest things to help though was the foundation of DORA (DevOps Research and Assessment) and the release of the DevOps Handbook. Both of these resources help provide the means of fundamental practices to follow for the ideal approach of Development and Operations. By showing the results of companies that implemented these policies and the cost savings that resulted, the initiative had solid proof of its effectiveness.

The entire purpose of the Lean Movement is to eliminate waste in product development. The result of this is increased efficiency and better delivery times. The idea is based on catching and fixing problems before they can rear their ugly head. By catching these issues early, adjustments can be made to keep the machine rolling and avoid stoppages. The Lean Movement is based on the following five principles:

1. Define a Value: Assigning value, good or bad, to vary aspects of a project both directly and indirectly.
2. Map the Value Stream: A visualization implemented of the value obtained in Principle 1.
3. Create a Flow: Avoid roadblocks. Determine how to prevent those that occur.
4. Establish Pull: Avoid overworking by creating a work queue so work can be started when time is available.
5. Seek Perfection: Continuous Improvement is the goal here. Analyze and make changes if something isn’t working. Looks for improvement opportunities.

When implementing these Principles into DevOps, we help improve upon what DevOps already provides. Collaboration and efficiency are improved as errors are caught early and the team isn’t overworked. The most important things are that these principles allow the team to quickly adapt to rising problems and maintain focus throughout the project. Continuous improvement is achieved as the team takes on a strong growth mindset that seeks constant ways to make the work process better for all.

The Agile Manifesto consists of 12 principles centered on the idea of continuous delivery. This delivery must be valuable to the customer and is delivered frequently. As a result of the frequent product delivery, feedback is received that can be used to implement new requirements for the product. Accept these to provide a great customer advantage. Motivated individuals are key and need to be supported with a proper environment and tools. Building on this, everyone must work together throughout the course of the project for best results. Increase overall effectiveness with regular meetings to work on efficiency opportunities in the workspace. Overall though, keep things simple, functioning, and always focus on good design and technical excellence. The best products are the result of attention to detail, patience, and effective collaboration. These principles combine to form the basis for Agile Development and eventually DevOps.

The Continuous Delivery Movement is based on the premise that segments of code, or pieces of the product, are prepared for release within the flow of development automatically. All changes to code created during the build stage are immediately sent to a testing environment. It is through this environment that the code is screened with automated tests to make sure the build is ready for deployment. This speeds up testing, delivery, and always ensures a proper deployment-ready build if properly implemented. The important thing here is that the testing covers multiple avenues to make sure the product is ready rather than testing simple parts. UI testing, Integration testing, load testing, and others are used to make sure the updates applied to the application are ready for the customer without issue. This greatly increases the thoroughness of each update by detecting issues before they can be deployed. Early bug detection results in faster fixes and less problems down the road. In turn, this produces a reliable update in shorter times and is ideal for Continuous Delivery and customer satisfaction.

Citations

*A Brief History of DevOps and Its Impact on Software Development*. (2023, February 26). EverythingDevOps. https://everythingdevops.dev/a-brief-history-of-devops-and-its-impact-on-software-development/

Atlassian. (n.d.). *Lean principles: advancing DevOps efficiency*. Atlassian. https://www.atlassian.com/agile/project-management/lean-principles

AWS. (n.d.). *What is Continuous Delivery? – Amazon Web Services*. Amazon Web Services, Inc. https://aws.amazon.com/devops/continuous-delivery/

IAN BUCHANAN. (n.d.). *History of DevOps*. Atlassian. https://www.atlassian.com/devops/what-is-devops/history-of-devops

*Principles behind the Agile Manifesto*. (2001). Agilemanifesto.org. https://agilemanifesto.org/principles.html